AN IMPROVED GRADIENT ALGORITHM FOR THE SOLUTION OF TWO-POINT BOUNDARY-VALUE PROBLEMS

JOHN ALLEN BATTENBURG

[redacted]

AN IMPROVED GRADIENT ALGORITHM FOR THE SOLUTION OF TWO-POINT BOUNDARY-VALUE PROBLEMS

by

John Allen Battenburg
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1959

Submitted in partial fulfillment of the
requirements for the degree of

ELECTRICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
June 1968

ABSTRACT

The gradient method for solving two-point boundary-value problems is discussed and a modification of the gradient direction is proposed. An algorithm for efficiently determining the step size is also derived. Analytic and numerical examples illustrating the efficiency of the method are presented.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

The author wishes to express his appreciation to Professor D. E. Kirk for his many hours of guidance and council, and to Professor H. A. Titus for his comments and suggestions.

1. THE PROBLEM AND NOTATION

The objective of controlling a physical system is to make the system function in the most desirable manner. If the system happens to be a production machine in a factory, the application of the best control will yield the maximum machine output for a specified expenditure of raw materials, power, and production time. Perhaps the system is an aircraft which is to be automatically controlled during landing. In this situation the best choice of control might be expected to minimize deviations from a specified flight path.

The system, or plant to be controlled, is assumed to be in the state variable form:

$$\dot{\underline{x}}(t) = \underline{a}(\underline{x}(t), \underline{u}(t), t) \tag{1}$$

where

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \text{ a } (n \times 1) \text{ vector} \qquad \underline{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \text{ a } (m \times 1) \text{ vector}$$

The effectiveness of a controlled process at accomplishing an assigned task is measured by a functional called the performance index or performance measure, $J(\underline{u})$. The performance measure is assumed to be of the form:

$$J(\underline{u}) = h\left[\underline{x}(t_f), t_f\right] + \int_{t_o}^{t_f} g(\underline{x}(t), \underline{u}(t), t)\, dt \tag{2}$$

The optimal control is the control which minimizes the performance measure.

There are two general approaches that can be followed when computing the optimal control. The first of these, dynamic programming, reduces the problem to one of making a finite number of optimal decisions starting at $t = t_f$ and working backwards in time. Decisions are based on the "principle of optimality" due to Bellman[1],[2] who has used the method extensively. While efficient, this method suffers from the requirement of large amounts of computer storage.

The alternative to dynamic programming lies in variational calculus. In this method, the performance measure is augmented by Lagrange multipliers and the functional

$$J_a(\underline{u}) = J(\underline{u}) + \int_{t_o}^{t_f} \underline{p}^T(t)(\underline{a}(\underline{x}(t),\underline{u}(t),t)-\dot{\underline{x}}(t))dt \qquad (3)$$

is obtained. The superscript T denotes the transpose of a vector or matrix. $\underline{p}$ denotes the costate (or adjoint) vector and is of dimension n x 1.

When the state equations are satisfied, as they must be, $J_a = J$ since the integrand of (3) vanishes.

Since a minimum of the augmented functional is sought, the variation of J_a (denoted δJ_a) is obtained and the fundamental theorem of variational calculus is employed to derive necessary conditions which must be satisfied by a candidate for an optimal solution.

It is convenient at this point to introduce a function, $\mathcal{H}$, the Hamiltonian, which is defined as:

$$\mathcal{H}(\underline{x}(t),\underline{u}(t),\underline{p}(t),t) \triangleq g(\underline{x}(t),\underline{u}(t),t)+\underline{p}^T(t)\left[\underline{a}(\underline{x}(t),\underline{u}(t),t\right] \tag{4}$$

Using this notation, the necessary conditions for optimality are: [See[3]]

$$\dot{\underline{x}}^*(t) = \nabla_{\underline{p}}\, \mathcal{H}(\underline{x}^*(t),\underline{u}^*(t),\underline{p}^*(t),t) \tag{5}$$

$$\dot{\underline{p}}^*(t) = -\nabla_{\underline{x}}\, \mathcal{H}(\underline{x}^*(t),\underline{u}^*(t),\underline{p}^*(t),t) \tag{6}$$

$$\underline{0} = \nabla_{\underline{u}}\, \mathcal{H}(\underline{x}^*(t),\underline{u}^*(t),\underline{p}^*(t),t) \tag{7}$$

and

$$\left[\nabla_{\underline{x}}\, h(\underline{x}^*(t_f),t_f) - \underline{p}^*(t_f)\right] \delta\underline{x}_F \tag{8}$$

$$+\left[\mathcal{H}(\underline{x}^*(t_f),\underline{u}^*(t_f),\underline{p}^*(t_f),t_f)+\frac{2}{2t}\, h(\underline{x}^*(t_f),t_f)\right]\delta t_f = 0$$

The star superscript denotes the optimal solution and $\nabla_{\underline{x}}\mathcal{H}$ denotes the matrix $\begin{bmatrix} \frac{\partial \mathcal{H}}{\partial x_1} \\ \vdots \\ \frac{\partial \mathcal{H}}{\partial x_n} \end{bmatrix}$

$\delta\underline{x}_F$ is the final variation of the state vector and δt_f is the variation of the final time.

The initial values of the states are assumed to be known and the relationships between the final conditions of the states and costates are given by equation (8). Thus the problem is a non linear two-point boundary-value problem.

Many proposals for solving this problem have been made to date. Among these methods are the gradient technique[4],[5],[6], variation of extremals[7], and the method of quasilinearization (generalized Newton-Raphson method)[8]. All of these methods involve an initial guess followed by an iterative procedure designed to calculate the optimal solution. Variation of extremals requires the designer to select initial values of the costate equations ($\underline{p}(t_o)$) and iterate until all final boundary conditions are satisfied. Quasilinearization demands an initial selection of the state and costate trajectories and the iterative process continues until the difference between successive trajectories becomes sufficiently small. Both of these methods may penalize the designer who makes an unfortunate initial guess by nonconvergence; however, convergence, when achieved, is swift. A general discussion of the gradient method is the topic of Section 2.

2. THE GRADIENT METHOD

In this section the gradient method is presented. The difficulties associated with the method, as well as its attributes, are discussed.

2.1 Description of the Gradient Method

Consider the system:

$$\dot{\underline{x}}(t) = \underline{a}(\underline{x}(t),\underline{u}(t),t) \tag{1}$$

and the performance measure:

$$J(\underline{u}) = h\left[\underline{x}(t_f),t_f\right] + \int_{t_o}^{t_f} g(\underline{x}(t),\underline{u}(t),t)\,dt \tag{2}$$

which is to be minimized.

The necessary condition for a solution to be optimal is that the variation $\delta J_a(\underline{u}^*)$ be zero, for problems with no constraints imposed on the control. Taking the variation of the augmented functional J_a and equating to zero results in the necessary conditions stated in equations (5),(6),(7), and (8).

The gradient method is started by selecting a trial control history $\underline{u}^{(i)}$ and satisfying equations (5),(6), and (8). The following equation then represents the entire performance measure variation.

$$\delta J\left(\underline{u}^{(i)}\right) = \int_{t_o}^{t_f} \nabla_{\underline{u}}^{T} \mathcal{H}\left(\underline{x}^{(i)}(t),\underline{u}^{(i)}(t),\underline{p}^{(i)}(t),t\right) \delta\underline{u}(t)\,dt \tag{9}$$

By choosing $\delta\underline{u}(t)$ correctly, a lower value of the perform-

ance measure can be found. The process is then repeated with the new trial control, $u^{(i+1)}$. This control is selected according to the relation:

$$\underline{u}^{(i+1)}(t) = \underline{u}^{(i)}(t) - \beta \nabla_{\underline{u}} \mathcal{H}(\underline{x}^{(i)}(t), \underline{u}^{(i)}(t), \underline{p}^{(i)}(t), t)$$

If the procedure converges on the nth iteration, the result is $\underline{u}^{(n)}(t) = \underline{u}^{*}(t)$.

Since the equations in all but a very few trivial problems are complicated (and usually non linear) and solutions in closed form are not readily available, the sets of differential equations are integrated numerically on a digital computer. Then $\underline{x}(t)$ and $\underline{p}(t)$ are not known for all values of t, but are available only at a discrete set of times, t_i, where $t_{i+1} = t_i + \Delta t$, with Δt being the integration step size. By similar arguments, the choice of control histories is restricted to some subset of U (the set of all admissible control histories) which is defined at the set of times (t_i) available during the computation.

A practical choice is to make $\underline{u}(t)$ a member of a set, called arbitrarily Ω, of all piecewise constant functions. That is $\underline{u}(t) = \underline{u}_i, t \epsilon \left[t_i, t_i + \Delta t \right]$.

All other classes of functions are approximated by Ω for the limiting case as $\Delta t \rightarrow 0$. The member of the set Ω chosen is denoted Ω_L, the subscript denoting the number of subintervals in the control interval $[t_o, t_f]$. The maximum number of piecewise constant controls in the control in-

terval is determined by the integration step size and is calculated from $L = \frac{t_f - t_o}{\Delta t}$.

$\delta J(\underline{u}^{(i)}) = 0$ implies either

a. $\nabla_{\underline{u}} \mathcal{H}\left(\underline{x}^{(i)}(t), \underline{u}^{(i)}(t), \underline{p}^{(i)}(t), t\right) \equiv \underline{0}, \quad t \epsilon \left[t_o, t_f\right]$

or

b. $\delta \underline{u}(t) \equiv \underline{0}$

or

c. $\int_{t_o}^{t_f} \left[\nabla_{\underline{u}}^{T} \mathcal{H}\left(\underline{x}^{(i)}(t), \underline{u}^{(i)}(t), \underline{p}^{(i)}(t), t\right) \delta \underline{u}(t)\right] dt = 0$

(10)

Case b above represents the trivial solution and is of no value. Of the non trivial choices, case a above is immediately appealing as it is identical to equation (7) of the set of necessary conditions. But, since $\underline{u}(t)$ is restricted to the set Ω_L (the price of numerical computation), $\delta J(\underline{u}^i)$ can only equal zero if the state and costate vectors also happen to be piecewise constant or if the optimal control happens to be piecewise constant. For this case there is no clear cut strategy to improve the control history and therefore proceed, eventually, to the optimal control. Present gradient techniques resort to satisfying the requirements dictated by equation (7) at the discrete points where all required information is available and use this data in the selection of a "better" control history.

This technique is satisfactory initially, but yields poor results as the optimum is approached.

Note should be made here that the gradient algorithm can only approach the true optimum as the integration step size Δt approaches zero. This is caused by demanding that $\underline{u}(t)$ be a member of the set Ω_L. It is not felt; however, that the error will be of great concern in problems of engineering interest.

2.2 Motivation for Selection of the Gradient Method

The gradient method, while suffering from difficulties yet to be discussed, has some appealing attributes.

The first favorable characteristic is the variable to be guessed, which in this method is the control history. The designer is much more likely to have insight into a suitable control history than either the remaining unknown boundary conditions required by variation of extremals or the trajectories required by quasilinearization. For a stable system, the initial guess at a control might well be to "do nothing" and guess $\underline{u}^{(0)}(t) = \underline{0}, t \epsilon [t_o, t_f]$. Equally important is the fact that the initial guess is not usually crucial to the success of the method.

Another feature of the gradient method is its relative ease of programming. For a stable system all integrations are numerically stable, as the stable state equations are integrated forward in time while the unstable costate equations are integrated backwards in time. All equations are relatively easy to derive. A flow diagram

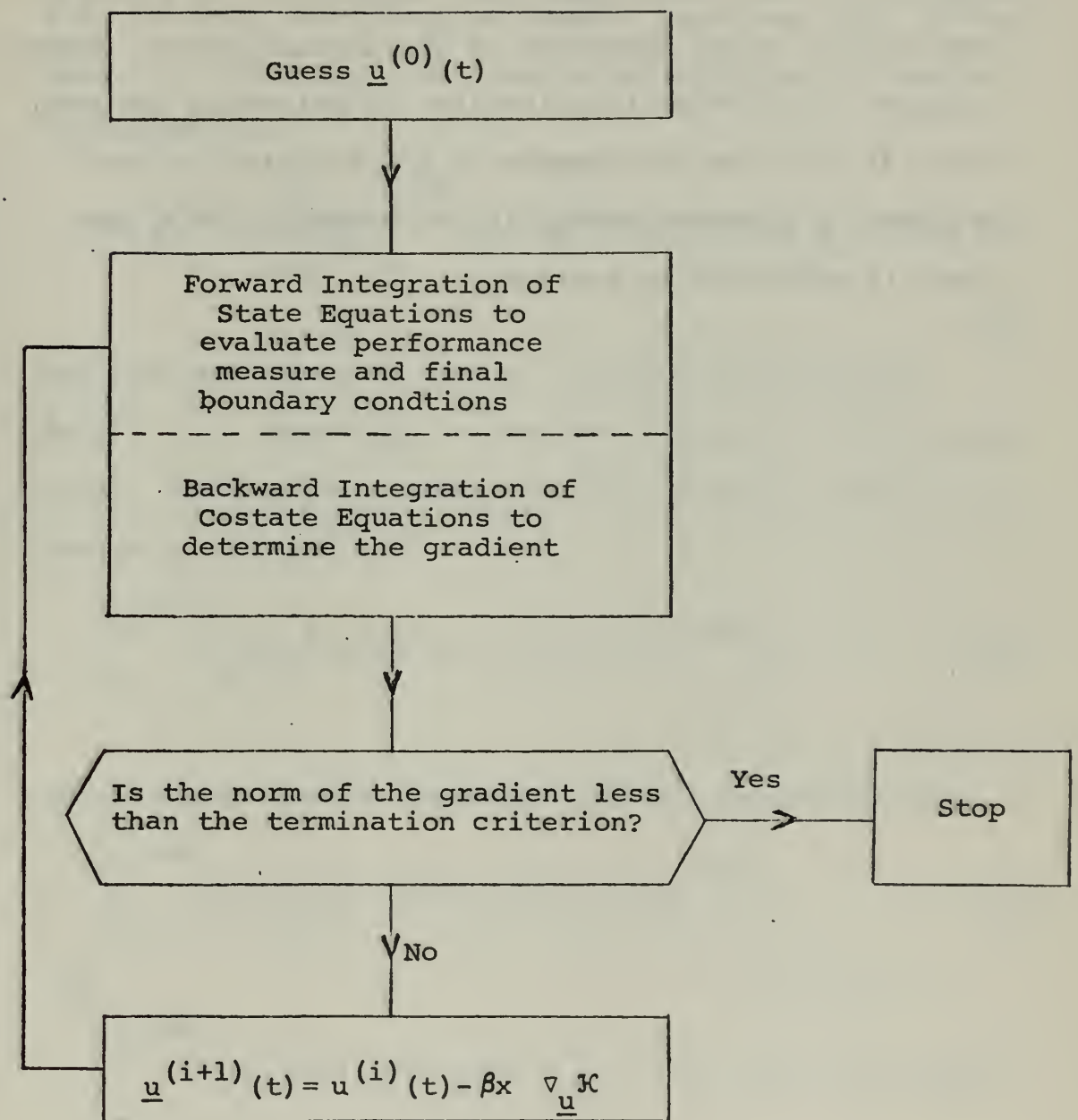

Figure 1.
Flow diagram of the Gradient Algorithm

of the algorithm is shown in Figure 1.

2.3 Difficulties Encountered with the Gradient Method

As mentioned above, the gradient method is plagued with some rather poor qualities, apparently the price of forgiveness in the selection of the initial guess. These problems are difficulties involved in selecting the step size (β) and slow convergence in the vicinity of the optimum. A proposed change in the selection of a gradient is presented in Section 3.

3. A MODIFICATION TO THE GRADIENT

An alternative solution to the dilemma of finding a strategy that will converge to the optimum choice of control (in the set Ω_L) is to restrict $\delta\underline{u}(t)$ to the set Ω_L also and base a strategy on case c (equation (10)) of section 2.1. With $\delta\underline{u}(t)$ so restricted equation (9) can be rewritten as:

$$\delta J\left(\underline{u}^{(i)}\right) = \sum_{j=0}^{L} \delta\underline{u}_j^T \int_{t_j}^{t_j+\Delta t} \nabla_{\underline{u}} \mathcal{H}\left(\underline{x}^{(i)}(t), \underline{u}^{(i)}(t), \underline{p}^{(i)}(t) t\right) dt \tag{11}$$

and the equivalent of case c is obtained by setting $\delta J(\underline{u}^*) = 0$. Since $\delta\underline{u}_j$ is not zero (except in the trivial case), the remaining condition for optimality can be immediately written as:

$$\int_{t_j}^{t_j+\Delta t} \nabla_{\underline{u}} \mathcal{H}\left(\underline{x}^*(t), \underline{u}^*(t), \underline{p}^*(t), t\right) dt \overset{\text{must}}{=} \underline{0} \qquad j = 0,1,2,\ldots,L \tag{12}$$

For a scalar control equation (12) may be expressed as:

$$\int_{t_j}^{t_j+\Delta t} \nabla_{u} \mathcal{H}\left(\underline{x}^*(t), u^*(t), \underline{p}^*(t), t\right) dt \overset{\text{must}}{=} 0 \qquad j = 0,1,2,\ldots,L \tag{13}$$

Let

$$\int_{t_j}^{t_j+\Delta t} \nabla_{u} \mathcal{H}\left(\underline{x}(t), u(t), \underline{p}(t), t\right) dt \triangleq F_j\left(\underline{x}(t), u(t), \underline{p}(t)\right)$$

With u restricted to the set Ω_L this functional may be written as $F_j = F_j(\underline{x}, u_1, u_2, \ldots, u_j, \ldots, u_L, \underline{p})$. From this

an expression for δu_j, the control variation in the jth interval is sought. An approximation to this variation may be obtained by expanding F_j about the reference trajectory.

$$F_j(\underline{x}+\delta\underline{x}, u_1+\delta u_1, u_2+\delta u_2, \ldots, u_j+\delta u_j, \ldots, u_L+\delta u_L, \underline{p}+\delta\underline{p})$$

$$= F_j(\underline{x}, u_1, u_2, \ldots, u_j, \ldots, u_L, \underline{p}) + \frac{\partial F_j}{\partial \underline{x}}\,\delta\underline{x} + \frac{\partial F_j}{\partial u_1}\,\delta u_1$$

$$+ \frac{\partial F_j}{\partial u_2}\,\delta u_2 + \cdots + \frac{\partial F_j}{\partial u_j}\,\delta u_j + \cdots + \frac{\partial F_j}{\partial u_L}\,\delta u_L + \frac{\partial F_j}{\partial \underline{p}}\,\delta\underline{p}$$

$$+ \text{higher order terms}$$

The terms containing first partial derivatives comprise the first variation of F_j and will be denoted δF_j. Of interest is the value of δF_j caused by the perturbation of the jth control.

Let $\delta u_i = 0,\ i \neq j$

Then

$$\delta F_j = \frac{\partial F_j}{\partial \underline{x}}\,\delta\underline{x} + \frac{\partial F_j}{\partial u_j}\,\delta u_j + \frac{\partial F_j}{\partial \underline{p}}\,\delta\underline{p}$$

Assuming that $\frac{\partial F_j}{\partial \underline{x}}\,\delta\underline{x}$ and $\frac{\partial F_j}{\partial \underline{p}}\,\delta\underline{p}$ are small enough to be neglected leads to a first order approximation of δF_j.

$$\delta F_j \doteq \frac{\partial F_j}{\partial u_j}\,\delta u_j$$

Then

$$F_j(\underline{x}+\delta\underline{x}, u_1, u_2, \ldots, u_j+\delta u_j, \ldots, u_L, \underline{p}+\delta\underline{p}) \approx$$

$$F_j(\underline{x}, u_1, u_2, \ldots, u_j, \ldots u_L, \underline{p}) + \frac{\partial F_j}{\partial u_j}\,\delta uj$$

For F_j in some neighborhood of zero, u_j must lie in some

neighborhood of u_j^* .

Let

$$F_j(\underline{x}+\delta\underline{x}, u_1, u_2, \ldots, u_j+\delta u_j, \ldots, u_L, \underline{p}+\delta\underline{p}) = F_j(\underline{x}^*, u_1, u_2, \ldots, u_j^*, \ldots, u_L, \underline{p}^*) = 0$$

Then

$$F_j(\underline{x}, u_1, u_2, \ldots, u_j, \ldots, u_L, \underline{p}) + \frac{\partial}{\partial u_j} F_j(\underline{x}, u_1, u_2, \ldots, u_j, \ldots, u_L, \underline{p}) \; \delta u_j = 0$$

and

$$\delta u_j = - \frac{F_j(\underline{x}, u_1, u_2, \ldots, u_j, \ldots, u_L, \underline{p})}{\frac{\partial}{\partial u_j} F_j(\underline{x}, u_1, u_2, \ldots, u_j, \ldots, u_L, \underline{p})} \tag{14}$$

If the control terms of the system state equations are no greater than quadratic and the control terms of the performance measure are quadratic, the denominator of equation (14) is a constant. For this case a nominal value of $\delta u_j = -F_j(\underline{x}, u_1, \ldots, u_L, \underline{p})/\Delta t$ may be employed.

In the notation of the problem this modified expression is written as:

$$\delta u_j = - \frac{1}{\Delta t} \int_{t_j}^{t_j+\Delta t} \nabla_u \mathcal{H}(\underline{x}(t), u(t), \underline{p}(t), t) \; dt \tag{15}$$

This expression indicates that an integral mean value strategy is employed to determine δu_j.

The time history, or collection of all the δu_j, is taken to be the gradient of $J(u^{(i)})$ with respect to the control, in the sense that adjusting the controls locally

in the direction of the gradient will result in the greatest change in the performance measure.

Use of this technique leads to a modified control gradient which is employed with the standard gradient algorithm. If the modification is valid, the resulting gradient should possess certain properties and eventually lead to the optimum choice of control. A general discussion of these properties is presented in Section 4.

4. A TEST OF METHOD EFFICIENCY

The analogy between the gradient method and a hill climbing procedure is presented. Two possible climbing techniques are stated, one of which generates orthogonal gradient vectors. A comparison of problems solved by the standard and improved gradient methods is made.

4.1 The Gradient Method as a Hill Climbing Procedure

The gradient method for the solution of optimal control problem is often compared to the hypothetical problem of a survey party attempting to climb to the top of a mountain (in our case descend to the floor of a valley) in a dense fog. It is desirable to reach the unknown and unseen summit in a minimum amount of time.

The head surveyor is faced with two likely strategies by which he may achieve the summit. He can take a local survey, move a few feet in the direction of steepest ascent, and then take another local survey to modify his direction. He may be reasonably assured that by use of this technique the shortest distance between his starting point and the top of the mountain will be covered. Unfortunately, a very high number of surveys will probably be necessary to maintain position on the path with the ensuing hazard of much "wasted" time.

An alternative to this strategy is to make a local survey and then proceed to climb in the indicated direction until no further ascent is possible. At this time another survey is made and the climb resumes in the new direction

determined. This approach is likely to deviate from the path of steepest ascent traversed before and runs the risk of running into plateaus which would give a false illusion of being at the summit. Its main virtue is the likelihood that fewer surveys will be required and, hopefully, the time of ascent will be diminished.

When following the second strategy, it is noticed that successive gradients will always be orthogonal. If this was not true, a further improvement could have been achieved by continuing to climb in the direction of the previous gradient.

The problem described above also occurs when using the gradient technique to compute an optimum control. The optimal control is the one which lies in the mathematical valley defined by the state equations of the system and the structure of the selected performance measure. Like the surveyor, the control designer can only observe local conditions and must use this data in his optimization procedure.

If a strategy similar to the alternative plan proposed above is followed, the computation of the optimal control proceeds as follows:

A control is selected and the gradient is calculated, establishing a direction of search. A number of trials are made to determine the minimum value of the performance measure that can be found by moving along this gradient. At the minimum value point a new gradient is computed and

the process is repeated. When no improvement can be attained in any direction, the process is assumed to have converged to a local minimum.

Using this type of procedure, if an accurate search for a minimum is performed, and if the gradient is correctly evaluated, successive gradients will be orthogonal.

4.2 Comparison of Standard and Modified Gradient Results

Problems computed using the standard gradient techniques approximate the orthogonality property in the initial stages, but as the optimum is approached any approximation to orthogonality completely breaks down. The process finally "converges" with an indication that, while the norm of the gradient is sufficiently large, no further improvement can be made by any move in the indicated gradient direction. This leads to the conclusion that the true gradient has not been computed from the standard techniques.

When the modified gradient is used with the same search procedure, successive gradients maintain the property of orthogonality throughout the problem computation and improvement is always achieved by moving along the indicated gradient. Convergence is smoother than that of the standard method and is usually achieved by the requirement that the norm of the gradient be less than some small specified constant. The conclusion is that the modification leads to the computation of an improved gradient of J with respect to the control $\underline{u}(t)$.

5. TWO SINGLE VARIABLE SEARCH PROCEDURES

Computation of the gradient establishes a direction in which a search for a minimum will prove fruitful. The size of the step to be made in this direction remains to be determined. Two single variable search procedures are presented by which this step size may be calculated.

5.1 The Golden Section Search

The Golden Section search described by Wilde [9] provides an efficient means for finding the minimum of an unknown, but assumed unimodal, function, once the minimum is known to lie in a specified closed interval. The principal properties of unimodality of interest are that for a unimodal function, $f(x)$, only a single minimum, $f(x^*)$, exists in a closed interval, I, and for two points, a and b, both in the interval, and on the same side of the minimum, $f(a) < f(b)$ if $|x^* - a| < |x^* - b|$.

The method is described briefly below.

Given: an interval of length L_0 containing a single minimum of $f(x)$

Find : an interval of length $L_1 < L_0$ such that the minimum of $f(x)$ is contained in L_1

A typical problem is shown in Figure 2. Let the lower bound of the interval be at a and the upper bound at b. Then the interval length L_0 is given by $L_0 = b-a$.

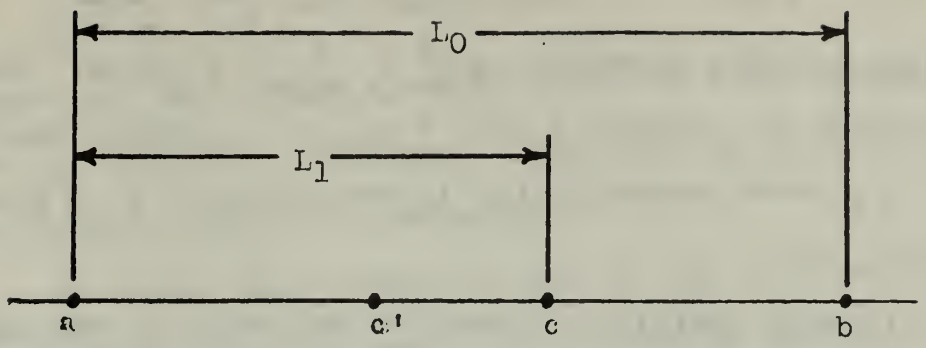

Figure 2.

Typical intervals in the Golden Section search algorithm

L_n/L_1	n
10^{-1}	5
10^{-2}	10
10^{-3}	15
10^{-4}	20
10^{-5}	25
10^{-6}	30
10^{-7}	34
10^{-8}	39

Table 1.

Interval reduction using the Golden Section search algorithm

Place an experiment in the interval at point c and let $L_1 = c-a$. Point c is chosen such that $\frac{L_o}{L_1} = \frac{L_1}{L_o-L_1}$. One solution of this equation is $\frac{L_o}{L_1} = \frac{2}{\sqrt{5}-1} = 1.68034$. Call this number τ. Then $c = a+L_1 = a+L_0/\tau$. Another experiment is placed symmetrically in the interval at a point $c' = b-L_0/\tau$.

The following decision is made on the basis of the experiments at points c and c':

if $f(c') < f(c)$ then $a < x^* < c$ and $L_1 = c-a$

if $f(c') = f(c)$ then $c' < x^* < c$ and $L_1 = c-c'$

if $f(c') > f(c)$ then $c' < x^* < c$ and $L_1 = b-c'$

Thus an interval $L_1 < L_0$ containing the minimum has been generated due to the unimodality of f(x) and the placement of the experiments.

The advantage of the method becomes apparent when further redcutions of the interval are attempted, the objective being to generate a sequence of decreasing interval lengths such that $L_n < \gamma$, where γ denotes the stopping criterion of the search.

To generate L_2, experiments are placed in the reduced interval L_1 in the same manner as the experiments were placed in the original interval. Assume that $a < x^* < c$ and thus $L_1 = c-a$. Two new experiments are to be placed in this new interval at points d and d'.

Then $d = a+L_1/\tau$ and $d' = c-L_1/\tau$

But $c' = b-L_0/\tau$ and, noting that $L_1 = L_0/\tau$, $d = a+L_0/\tau^2$.

An expression for c'-d may be written as:

$$c'-d = L_o\left(1 - \frac{1}{\tau} - \frac{1}{\tau^2}\right) = 0 \quad \therefore c' = d$$

Since one experiment already exists in the reduced interval L_1, only one additional experiment must be performed to further reduce the interval.

To apply this search method to the gradient algorithm the interval of the search must first be defined. It is already known that moving in the gradient direction will guarantee at least some improvement in the performance measure. A crude search is performed until the performance measure is greater than the value found at the origin of the search. This is achieved by initially taking a unit step in the gradient direction and doubling subsequent step sizes until this condition is met. The resultant closed interval must then contain the minimum and can be reduced by means of the golden section search procedure. Table 1 shows the number of experiments required to reduce an interval of unit length to a desired length L_n.

Application of the golden section search and modified gradient technique to the problems run to date has resulted in convergence in a very few iterations and established the validity of the modified gradient method. Many experiments are required per iteration, however; and so the total number of forward integrations that must be performed is still rather high. A method to improve this shortcoming is discussed next.

5.2 A Quadratic Approximation Search Technique

Since the search for a minimum in the gradient direction involves the single variable β, J can be written as some unknown function of this variable $J(\beta)$. From previous calculations $J(0)$ is known and $\frac{d}{d\beta} J(0)$ is known to be less than or equal to zero, with equality applying at the optimum solution of the main problem.

An alternative solution to performing a golden section search is to, rather boldly, assume that $J(\beta)$ is a quadratic of the form $J(\beta) = a\beta^2+b\beta+c$ and then to use the minimum of this quadratic as the optimum value of β. While there is no reason to believe that this should be as accurate as a search procedure, it requires far fewer experiments to achieve an estimate.

Given a quadratic, $f(x)$, and three points $x_1 < x_2 < x_3$, it can be shown that if $f(x_1) > f(x_2)$ and $f(x_3) > f(x_2)$, then $x_1 < x^* < x_3$ where $f(x^*) = \min_x f(x)$.

A quadratic may be fitted exactly to any three points x_1, x_2, and x_3 in the following manner.

Let $f(x) = ax^2+bx+c$ where a, b, and c are unknown constants.

Using the values of $f(x)$ at the three known points the following equations may be written:

$$f(x_1) = ax_1^2+bx_1+c$$

$$f(x_2) = ax_2^2+bx_2+c$$

$$f(x_3) = ax_3^2+bx_3+c$$

or in matrix notation

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

The expression is easily solved for the vector of coefficients.

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{bmatrix}^{-1} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

The matrix $\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{bmatrix}$ is positive definite and therefore its inverse always exists. Thus the coefficients a, b, and c can always be computed.

The minimum of this quadratic is found by setting the derivative equal to zero. This results in $x^* = -b/2a$.

The three test points through which the quadratic is to be fitted are again found rather crudely. The first point is placed at the origin of the search. No experiment need be performed here as the value of the performance measure is already known. A test is next performed with a step size of unit length and the results of this test determine the direction of further search along the gradient.

1. If the performance measure at this point is greater than that at the origin the third test point is placed midway between the preceding test and the origin. Successive tests are made in this manner until the value of the performance measure is less than that found at the origin, and finally, the quadratic form is fitted to the three points closest to the origin.
2. If the performance measure at the first test point is less than that at the origin, the step size is doubled and further tests are placed in this manner until the performance measure increases in value. The quadratic is then fitted through the three test points that are farthest from the origin.

Once the three points have been determined, the constants a and b are easily evaluated and the quadratic approximation of β^* generated.

6. AN ANALYTIC EXAMPLE

A simple example will serve to illustrate some of the differences between the standard gradient method and the method utilizing the modified gradient computation.

Consider the linear first order system:

$$\dot{x}(t) = -x(t) + u(t)$$

and the performance measure:

$$J = x^2(t_f) = \int_{t_o}^{t_f} 1/2 u^2(t)\, dt$$

The following additional data is given.

$$t_o = 0,\ t_f = 1,\ x(t_0) = x_0 = 4.0,\ x(t_f) \text{ unspecified}$$

Forming J_a and taking the variation, the necessary conditions can be written.

1. $\dot{x}^*(t) = -x^*(t) + u^*(t)$ with B.C. $x^*(t_0) = x_0$
2. $\dot{p}^*(t) = p^*(t)$
3. $p^*(t_f) = 2x^*(t_f)$
4. $u^*(t) = -p^*(t)$

These equations may be solved simultaneously for the optimal solution.

$$x^*(t) = x_0 e^{-t} - k/2 \left[e^t - e^{-t}\right]$$

$$p^*(t) = ke^t,\ p^*(1) = 2x^*(1)$$

$$u^*(t) = -ke^t$$

with $x_0 = 4.0$

$k = 0.58063$, $u^*(t) = -0.58063e^t$

$x^*(1) = 0.78915$, $J^* = 1.16126214$

This will be considered as the reference solution.

To exaggerate the difference between the two methods, the best control in the set Ω_1 will be calculated. With this choice of control, the following relationships are found.

$$x(t) = x_0e^{-t}+u(1-e^{-t})$$

$$p(t) = ke^t, \; k = 2e^{-tf}x(t_f)$$

$$\nabla_u \mathcal{H}(x(t),u(t),p(t),t) = u+ke^t$$

The procedure is started by selecting an initial control history and choosing the next control according to the method selected. If the integration is done numerically on a digital computer with a step size of 1.0, the only values of x and p available are those at the end points of the interval. The initial guess of u(t) will be 0.0.

Case I. Standard gradient procedures

δu is chosen to satisfy equation (7) at discrete points in the interval. For an extreme case δu is based on conditions existing at t = 0.

Then $\delta u^{(i)} = -u^{(i)}-k$

and $u^{(i+1)} = u^{(i)}+\delta u^{(i)}$

Applying this to the problem equations:

$$x(1) = x_0 e^{-1} + u^{(i)}(1-e^{-1})$$

$$k = 2e^{-1}x(1)$$

$$u^{(i+1)} = -k$$

Elimination of variables results in a difference equation in u.

$$u^{(i+1)} = -0.271x_0 - 0.469u^{(i)}$$

which converges to $u^* = -0.1845x_0$.

Application of this control to the system results in

$$u^* = -0.739, \quad x^*(1) = 1.006, \quad J^* = 1.285$$

note that: $u(t) + p(t)\big|_{t=0} = -0.739+0.739 = 0$

Apparently the "gradient" is zero and the optimal control has been found.

However, a direct search among controls in the set Ω_1 results in

$$u^* = -1.034, \quad x^*(1) = 0.817, \quad J^* = 1.203$$

obviously a better choice of control than that generated by the standard gradient method.

Case II. Modified gradient method

δu is chosen to satisfy

$$t_f - t_o \int_{t_o}^{t_f} \left[u(t)+p(t)\right] dt + \delta u^{(i)} = 0$$

which is the integral expression in equation (15).

Then $\delta u^{(i)} = -\frac{1}{\Delta t}\int_{t_o}^{t_f}\left[u^{(i)}(t)+p^{(i)}(t)\right]dt$

where $\Delta t = t_f - t_o$

$$= -\frac{1}{\Delta t}\left[u^{(i)}\Delta t + k(e^1-1)\right]$$

and

$$u^{(i+1)} = -\frac{k}{\Delta t}(e^1-1)$$

After application to the problem equation set and manipulation, a new difference equation in u is obtained.

$$u^{(i+1)} = -0.465x_0 - 0.8u^{(i)}$$

with solution

$$u^* = -0.258x_0$$

Application of this control to the system results in

$$u^* = -1.034, \quad x^*(1) = 0.818, \quad J^* = 1.203$$

This is virtually the same result as found by direct search.

Of the choices offered by the two methods, the modified gradient (which is scalar for this problem) converged to the optimum solution for this type of control($u(t) \epsilon \Omega_1$).

The control gradient for this problem is sketched in Figure 3. which shows conditions existing after the optimum control has been found. Examination of this figure reveals that this is <u>not</u> the optimum solution when all admissible choices of control are considered. A suitable choice for an admissible $\delta u(t)$ is shown in the figure.

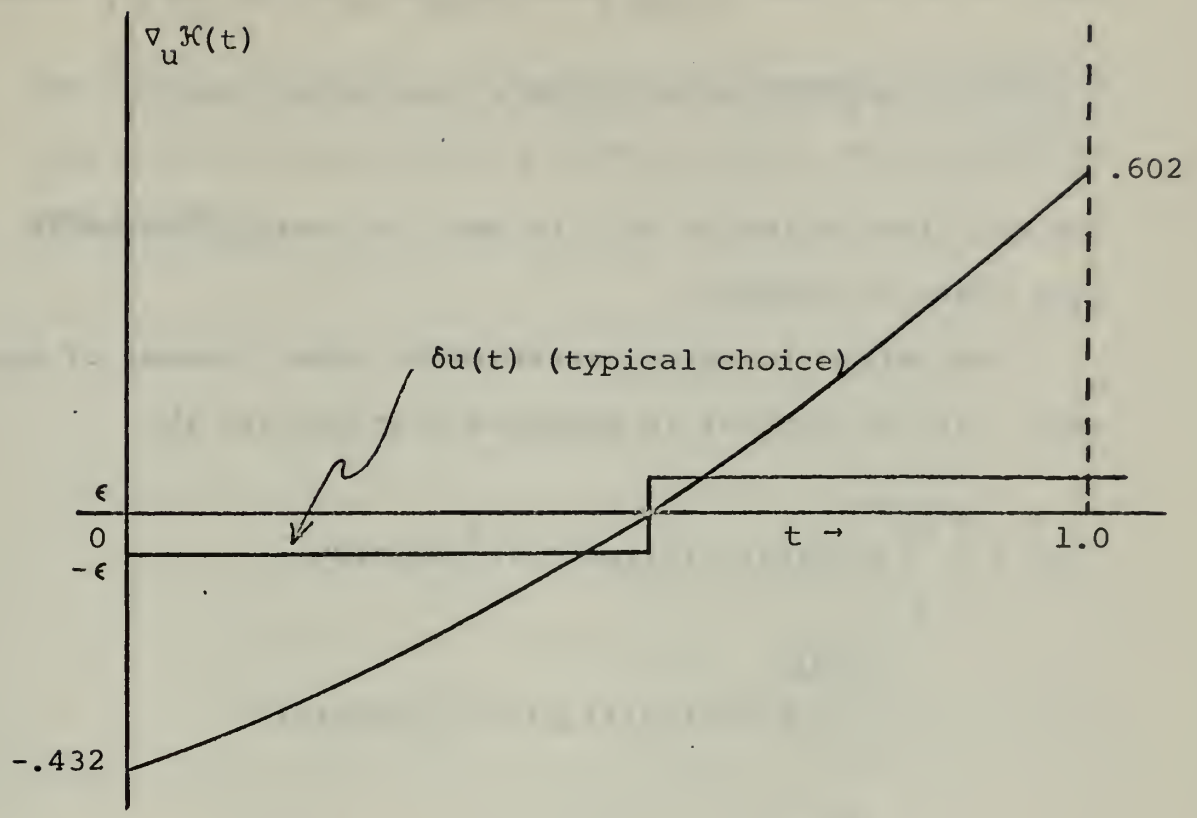

Figure 3

Linear Regulator - $\nabla_u \mathcal{H}(t)$ vs. time at optimal solution for class Ω_1 control. An admissible control variation is shown.

The resulting δJ is 0.256ϵ.

What we have found is the best control in the set Ω_1 since,

$$\int_{t_o}^{t_f}\left[u^*(t)+ p^*(t)\right]dt = -1.034 + .602\left[e^1-1\right] = 0$$

and thus the variation in J is zero for small changes in this class of control.

The method is simply extended to other classes of control. If the control is selected from the set Ω_2.

$$\delta J = \int_{t_o}^{t_o+\Delta t}\left[\nabla_u \mathcal{H}(x(t),u(t),p(t),t)\right]\delta u(t)dt$$

$$+ \int_{t_1}^{t_1+\Delta t}\left[\nabla_u \mathcal{H}(x(t),u(t),p(t),t)\right]\delta u(t)dt$$

here $\Delta t = \dfrac{(t_f-t_o)}{2}, t_{j+1} = t_j+\Delta t,\ j = 0,1$

$$t_o = 0$$

$$\delta u(t)\ \epsilon[\Omega_2]$$

and $$\delta u_j = -\frac{1}{\Delta t}\int_{t_j}^{t_j+\Delta t}\left[\nabla_u \mathcal{H}(x(t),u(t),p(t),t)\right]dt,\quad j = 0,1$$

When applied to the example problem, a set of difference equations is obtained.

$$u_1^{(i+1)} = -0.352x_0 - 0.281u_1^{(i)} - 0.385u_2^{(i)}$$

$$u_2^{(i+1)} = -0.583x_0 - 0.360u_1^{(i)} - 0.642u_2^{(i)}$$

The solution of these equations is:

$$u_1^* = -0.757,\ u_2^* = -1.255,\quad \text{for } x_0 = 4.0$$

When these controls are applied to the regulator the results are:

$$x^*(1) = 0.795, \text{ and } J^* = 1.169$$

The optimal solution presented above compares closely to the solution attained by a direct search of controls in the set Ω_2.

7. COMPUTER PROGRAM AND NUMERICAL RESULTS

An explanation of the computational program is presented. The section concludes with the results of problems computed using the standard and modified gradients.

7.1 The Gradient Program

A flow diagram of the gradient algorithm was presented in section 2. This formulation was followed in coding the computer programs following. The main program accomplishes all "bookkeeping" operations required and contains logic statements for problem termination. Various subroutines are called to perform computations and other auxiliary functions. The interconnection of these programs is shown in Figure 4.

The principal subsections of the main program and the subroutines are as follows:

A. The input section

The following inputs are required.

1. The state initial condition vector $-\underline{x}_o$
2. The initial and final times
3. The allowable integration step size for numerical accuracy
4. The number of control intervals in the problem
5. The convergence criterion

B. Problem initialization

This section divides the problem time into the desired number of equal control intervals and

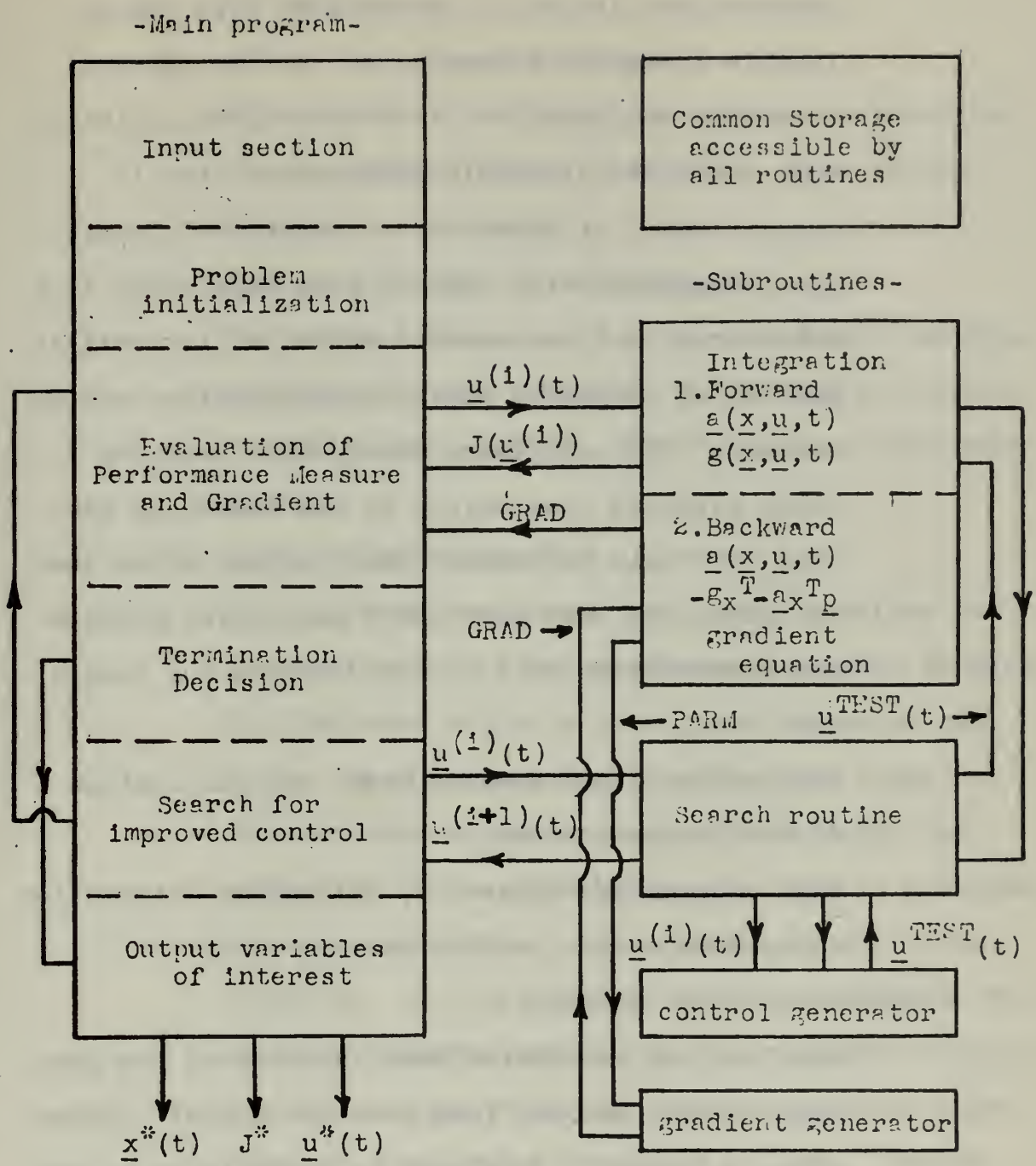

Figure 4.
Gradient program
Interconnection diagram

compares the length of an interval with the allowable integration step size. If the interval length is too great, it is divided into subintervals until the allowable integration step is greater than, or equal to, a subinterval length. The integration step size is then made equal to a subinterval and the control during an interval is made up of identical controls in contained subintervals. This maintains integration accuracy while allowing flexibility in the number of control intervals selected. Other values which need be calculated only once for a particular problem are generated at this time and stored for future use.

C. Evaluation of performance index and calculation of the gradient vector

This process is achieved by calling an integration subroutine with a control history u(t).

D. Termination decision

Problems are terminated when the norm of the gradient vector becomes less than the preset convergence criterion. This norm is taken to be of the form

$$\|\nabla_u \mathcal{H}\| = \frac{1}{L} \sum_{i=0}^{L-1} \left[\frac{1}{\Delta t} \int_{t_i}^{t_i+\Delta t} \nabla_u \mathcal{H}(\underline{x}(t), u(t), \underline{p}(t), t)\, dt \right]^2$$

where L is the number of control intervals.

An alternate exit is made after a preset number of iterations.

E. Search for improved control

This search is performed by calling a subroutine of the desired search algorithm. This section returns to section C.

F. Output variables of interest

The control history and up to four state variable trajectories are tabulated and graphs of desired variables are plotted. This completes the computation.

G. Integration subroutine

This routine integrates the state equations and integral terms of the performance measure forward in time when called with a given choice of control. At the final time non integral terms of the performance measure are evaluated and the cost ($J(\underline{u}^{(i)})$ is calculated. An exit is provided at this point when a new gradient is not to be computed. If the gradient is to be computed, the costate boundary conditions are evaluated ($\underline{p}(t_f)$), and the states, costates, and the gradient equation are integrated backwards in time. During the integration, the integral mean value of the gradient is computed for each control interval and stored. When the initial time is reached, the gradient generator is called with the array of

mean values and exit is made to the calling program.

H. Search subroutine

This routine computes a new trial control $u^{(i+1)}(t)$ by placing experiments at a distance β along the gradient vector. An experimental control is generated by choosing a value of β and calling the control generator. An experiment is made by calling the integration subroutine with this experimental control and evaluating the performance measure. The best distance (β^*) is computed according to whichever algorithm (golden section search or quadratic approximation) is selected. The new control is calculated by the control generator and exit is made to the calling program.

I. Gradient generator

When called with a vector of parameters whose dimension is equal to the number of control intervals, this subroutine generates a new vector whose dimension is the number of integration steps in the problem. All values of the newly generated vector in a control interval have the same value as the single parameter associated with that interval.

J. Control generator

This routine is called with a reference control, a reference gradient and a scalar multiplier. The

new control generated is the sum of the reference control and the product of the scalar and the gradient.

This same program is modified to compute a solution according to the standard gradient technique by evaluating the gradient equation at the beginning of each control interval and using this vector as the gradient.

7.2 A Linear Regulator

The program was initially applied to the linear regulator problem of section 6. An integration step size of 0.01 and an initial guess of control, $u^{(o)}(t) = 0.0$ were used. The control history was selected from Ω_{100}. A golden section search with a convergence interval of $1x10^{-5}$ was employed to ensure accuracy in determining the best step sixe. The convergence criterion for the gradient norm was $1x10^{-6}$.

Both methods converged on the second iteration to a value of 1.16126633 for the performance measure. Results of the problems are shown in Table 2.

The fortunate initial guess of control and the structure of the problem enable both methods to achieve convergence after only two iterations. Of interest is the primary indicator of convergence, the norm of the gradient vector, which has indicated convergence with the modified method but has failed to achieve even the same order of magnitude using standard techniques. Both of these computations incorporated an exit when the difference between

Iteration	Cost	Gradient Norm	Gradient Angular Change	β^*
Standard Gradient				
0	2.16536427	370.75071539	86.8°	0.538821
1	1.16126633	0.00211787	179.9°	0.072949
2	1.16126633	0.00158204	--	--
Modified Gradient				
0	2.16536427	374.45974144	86.8°	0.536446
1	1.16126633	0.00003283	179.9°	0.499995
2	1.16126633	0.00000015	--	--

Table 2

Linear Regulator - Computation History

$u^{(0)}(t) = 0.0$

successive costs was less than 1×10^{-6}.

Another computation was made with this cost exit removed and an initial guess of -1.0 for the control. A maximum of 20 iterations was permitted. The results of this test are shown in Table 3.

While the initial guess of -1.0 for control is closer to being optimal than the guess of 0.0 used previously, both methods are in a poorer starting position computationally. The standard gradient technique fails to achieve convergence after 20 iterations (the maximum allowed). Of primary interest are the gradient angular change and the step size, β^*. While the standard gradient is roughly orthogonal initially, as the optimum is approached successive gradients are very nearly identical; moreover, the step size drops to the minimum value the search algorithm can generate. A gradient supposedly exists but search along this gradient fails to disclose a better control.

The modified gradient computation, on the other hand, indicates a rather smooth convergence to the gradient norm cutoff and the orthogonality of successive gradients is maintained throughout the computation. The step size, β^*, exhibits a tendency to form a pattern and, more importantly, improves the control at each iteration. It is concluded, therefore, that the method has calculated the gradient accurately.

The optimal value of the performance measure generated by the modified method compares favorably with the true

Iteration	Cost	Gradient Norm	Gradient Angular Change	B^*
Standard Gradient				
0	1.20458767	9.45135627	97.1°	0.860359
1	1.16360526	0.82419079	91.4°	0.855713
2	1.16142950	0.02993304	74.2°	1.025390
3	1.16126862	0.00065597	109.0°	0.000005
4	1.16126862	0.00065596	0.1°	0.000005
5	1.16126862	0.00065594	0.1°	0.000005
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
20	1.16126862	0.00065577	--	--
Modified Gradient				
0	1.20458767	9.60531177	97.0°	0.845526
1	1.16357991	0.85556082	90.0°	0.594545
2	1.16143656	0.03768012	90.0°	0.845523
3	1.16127727	0.00355629	90.0°	0.594538
4	1.16126729	0.00014781	90.0°	0.845523
5	1.16126667	0.00001316	90.0°	0.594478
6	1.16126663	0.00000058	--	--

Table 3

Linear Regulator - Computation History

$u^{(0)}(t) = -1.0$

optimal value of 1.16126214 computed by digital evaluation of the analytic solution of the problem. The error arises from the piecewise constant control restriction.

7.3 Continuous Stirred Tank Reactor

A more ambitious undertaking is the solution of the continuous stirred tank reactor problem posed by Lapidus and Luus[10].

The state equations of the system are:

$$\dot{x}_1(t) = -2(x_1(t)+0.25)+(x_2(t)+0.5)\exp\left(\frac{25x_1(t)}{x_1(t)+2}\right) - (x_1(t)+0.25)u(t)$$

$$\dot{x}_2(t) = 0.5 - x_2(t) - (x_2(t)+0.5)\exp\left(\frac{25x_1(t)}{x_1(t)+2}\right)$$

and the performance measure to be minimized is:

$$J = \int_{t_o}^{t_f} \left[x_1^2(t) + x_2^2(t) + 0.1u^2(t) \right] dt$$

Forming the augmented performance measure and utilizing the necessary conditions results in the formulation of the costate equations and gradient equation.

$$\dot{p}_1(t) = -2x_1(t)+2p_1(t) - p_1(t)(x_2(t)+0.5)\left[\frac{50}{(x_1(t)+2)^2}\right]\exp\left(\frac{25x_1(t)}{x_1(t)+2}\right) + p_1(t)u(t)+p_2(t)(x_2(t)+0.5)\left[\frac{50}{(x_1(t)+2)^2}\right]\exp\left(\frac{25x_1(t)}{x_1(t)+2}\right)$$

$$\dot{p}_2(t) = -2x_2(t) - p_1(t)\exp\left(\frac{25x_1(t)}{x_1(t)+2}\right) + p_2(t)\left[1+\exp\left(\frac{25x_1(t)}{x_1(t)+2}\right)\right]$$

$$p_1(t_f) = p_2(t_f) = 0$$

$$\nabla_u \mathcal{H}(\underline{x}(t), u(t), \underline{p}(t), t) = 0.2u(t) - p_1(t)\left[x_1(t) + 0.25\right]$$

The other given data are:

$t_0 = 0.0$, $t_f = 0.78$

$x_{10} = 0.05$, $x_{20} = 0.0$, $\underline{x}(t_f)$ unspecified

The control to be generated is the flow of coolant through a coil immersed in the reactor. $x_1(t)$ is the deviation from the desired steady state temperature and $x_2(t)$ is the deviation from the desired steady state chemical concentration in the reactor.

Solution of this problem using variation of extremals indicates the optimum value of the performance measure to be .02660336. As in the linear regulator problem, the initial investigation is a comparison between the standard gradient method and the modified technique. The step size is determined by a golden section search.

Three exits were provided

1. Norm of gradient less than $1x10^{-7}$
2. Improvement between successive iterations less than $1x10^{-6}$
3. 20 iterations have been completed

The integration step size selected was 0.01 and 78 control intervals were provided. $u^{(0)}(t)$ was chosen as

0.0. The results of the investigation are shown in Table 4.

From this data, the principal advantage of the modified gradient method is the orthogonality maintained by successive gradients, reinforcing the earlier conclusion that this method is computing an improved control gradient. Both methods terminated when the difference between successive evaluations of the performance measure was sufficiently small, but the standard gradient terminated more slowly and to a greater value.

With this exit removed, the modified gradient converged to the desired gradient norm on the ninth iteration. The value of J computed was 0.2660936. Successive gradients maintained the orthogonality property throughout the process.

Under the same conditions, the standard gradient method failed to converge after 20 iterations. Successive gradients became less and less orthogonal as the optimum was approached and the algorithm began to break down after 16 iterations. The minimum value of J computed was 0.0266120l.

Some further insight into the convergence tendencies of the two methods is gained by graphically comparing the logarithms of the gradient norms and the normalized errors in the performance measure. This latter quantity is generated as $\epsilon^{(i)} = \frac{J^{(i)} - J^*}{J^*}$, where J^* is the value computed using variation of extremals. These results are shown in Figures 5 and 6 respectively.

The modified gradient technique converges smoothly to

Iteration	Cost	$\log_{10}$ Gradient Norm	Gradient Angular Change	β^*
Standard Gradient				
0	0.22583207	-1.03914642	106.4°	1.623640
1	0.02729169	-3.45231247	81.3°	1.956198
2	0.02697532	-2.98352318	88.9°	0.302558
3	0.02677793	-3.95951080	104.8°	0.656046
4	0.02674786	-3.80315587	83.8°	0.408785
5	0.02670983	-3.95664883	108.2°	0.295981
6	0.02669719	-4.31456280	72.9°	1.216354
7	0.02666300	-3.71768951	94.5°	0.190202
8	0.02664876	-4.68956852	85.3°	1.565506
9	0.02663578	-4.23825169	84.2°	0.343573
10	0.02662267	-4.69813728	124.2°	0.155175
11	0.02662192	-5.14795015	--	--
Modified Gradient				
0	0.22583207	-1.05710602	106.4°	1.657650
1	0.02729702	-3.48081589	90.0°	3.500311
2	0.02672917	-3.12831785	90.0°	0.256890
3	0.02662924	-5.04592037	90.0°	3.586506
4	0.02661515	-4.58679676	90.0°	0.229326
5	0.02661015	-6.45678598	90.0°	3.644995
6	0.02660953	-5.92702579	--	--

Table 4

Stirred Tank Reactor - Computation History

$u^{(0)}(t) = 0.0$

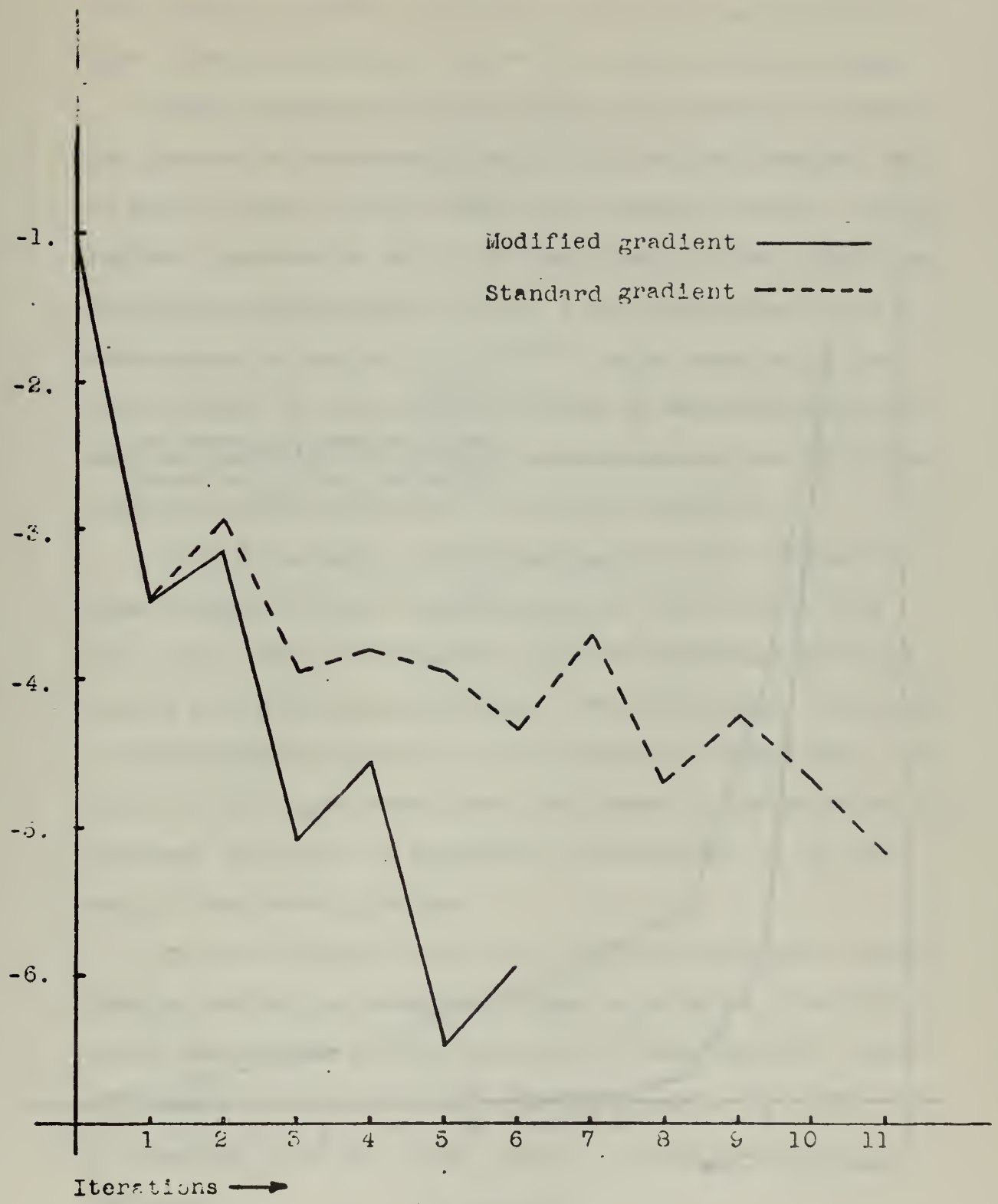

Figure 5

Stirred Tank Reactor - Log_{10} Gradient Norm vs. Iterations

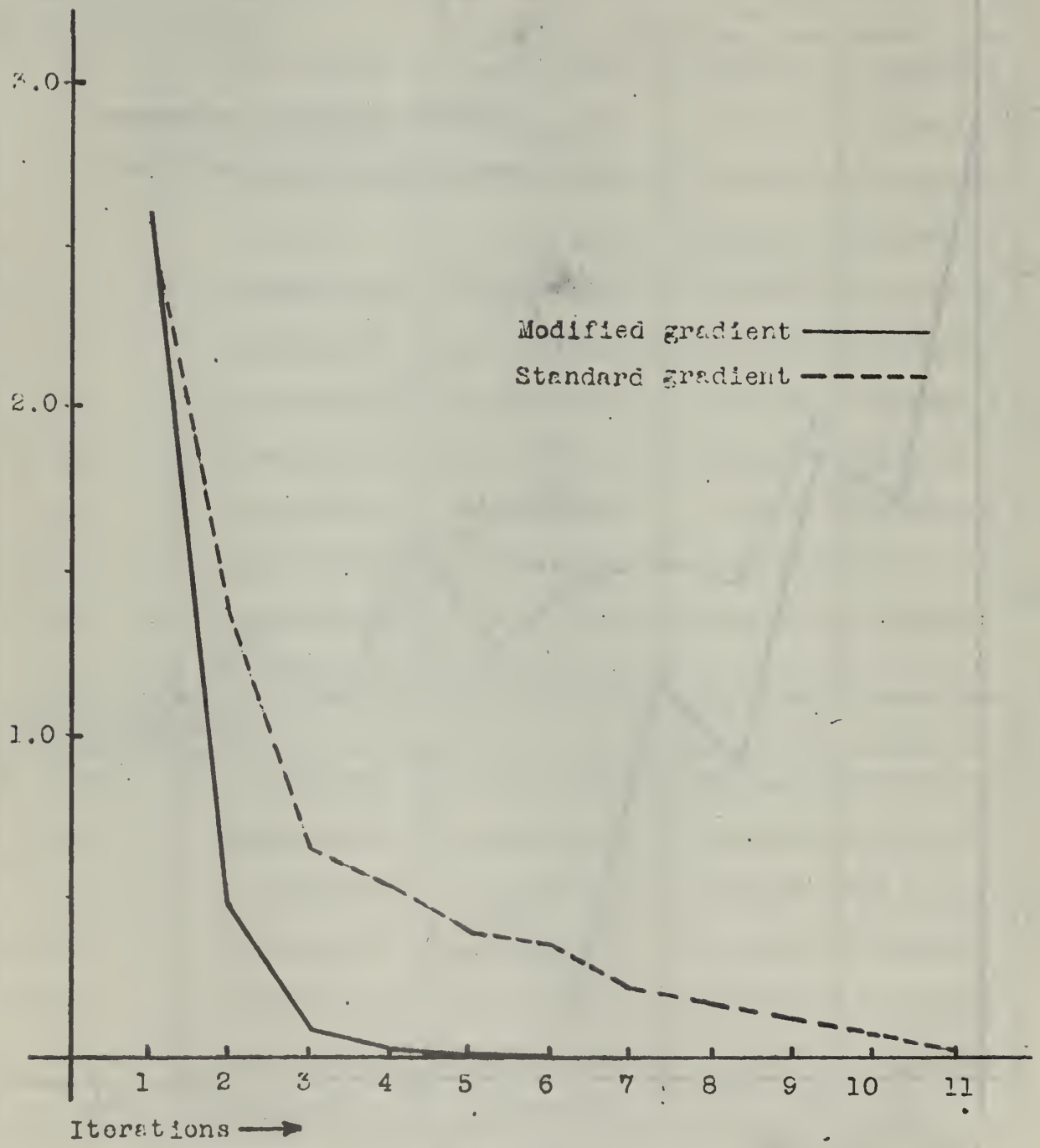

Figure 6

Stirred Tank Reactor - Percent Cost Error vs. Iterations

the optimum value for the class of control while the standard technique behaves somewhat erratically and converges more slowly once values "near" the optimum are attained.

The logarithm of the gradient norm exhibits a definite pattern of convergence with the modified gradient and is again rather erratic under the standard method. In a further computation using the modified gradient technique, the pattern displayed in Figure 5 was maintained up to a convergence criterion of $1x10^{-13}$. This computation converged after 19 iterations although no measurable improvement in the value of the performance measure was attained beyond the ninth iteration value of 0.02660936.

All of the above investigations involved golden section searches in the determination of the correct step size and, while convergence using the modified gradient method is achieved in relatively few iterations, the number of experiments performed in the search is rather high. A total of 195 experiments were performed by the modified gradient technique in computing the solution of the presently considered problem.

The next investigation utilized the quadratic approximation method to determine a step size which gives efficient improvement with relatively few experiments. While the number of iterations may be expected to increase, it is expected that the total number of experiments will be decreased.

A comparison between this method and a golden section

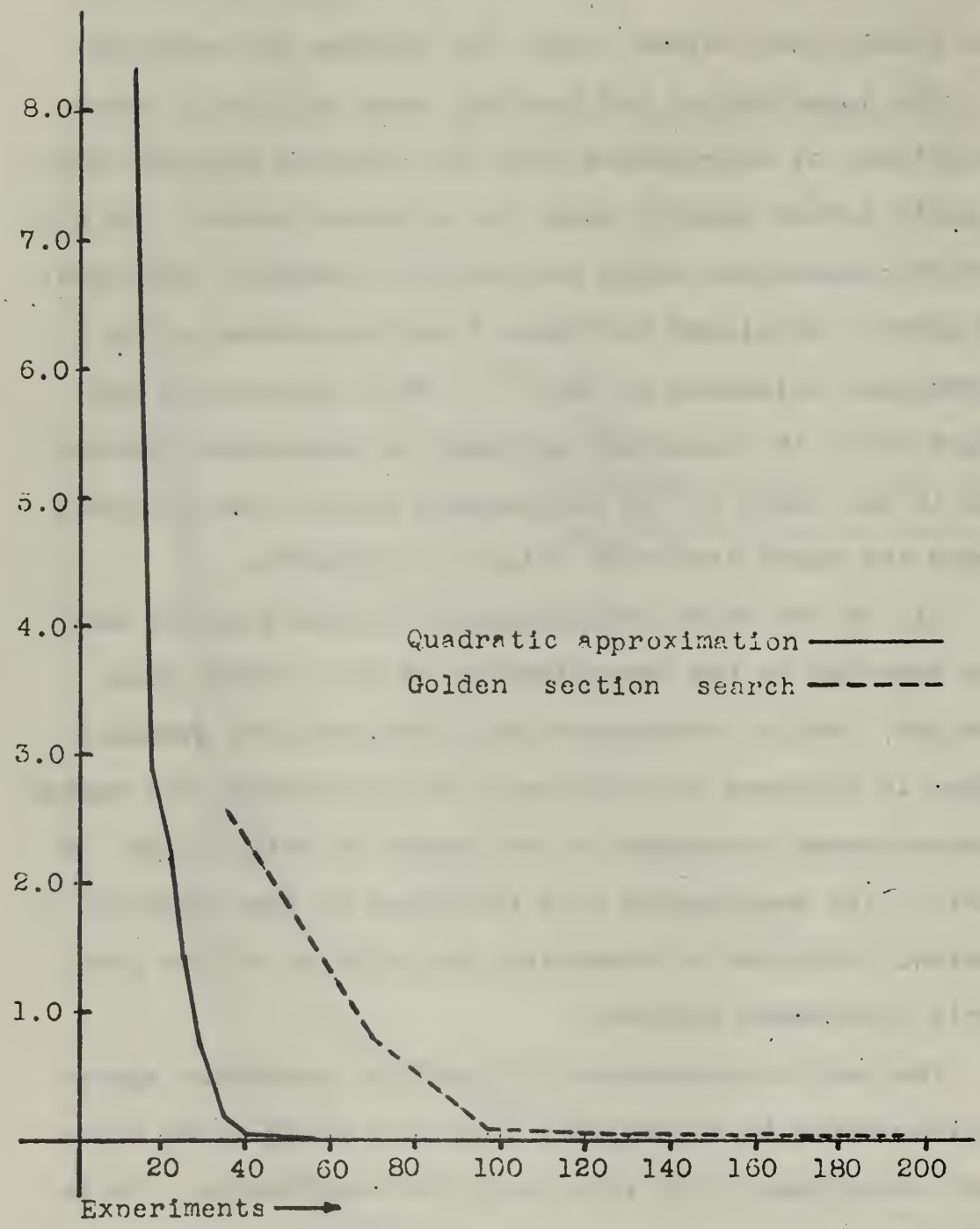

Figure 7

Stirred Tank Reactor - Percent Cost Error vs. Experiments

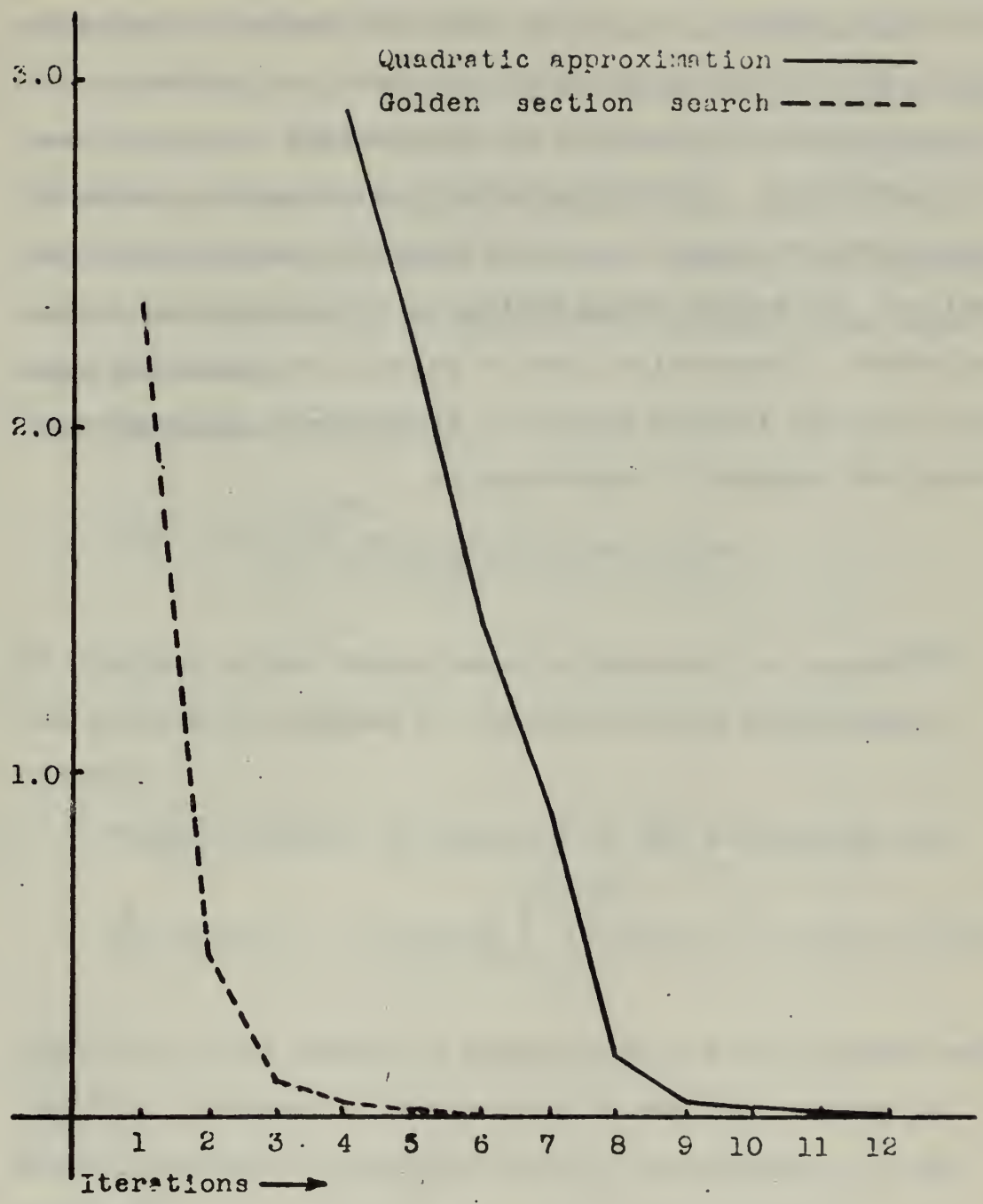

Figure 8

Stirred Tank Reactor - Percent Cost Error vs. Iterations

search is shown in Figures 7 and 8.

The number of experiments made in the computation is markedly reduced (to 57) by using the quadratic approximation, and, while poorer initially than the golden section, compares quite favorably with that method as the optimum is approached. Successive gradients using the quadratic approximation behave erratically during the initial iterations, but become nearly orthogonal as the optimum is approached. Computation time is sharply decreased as expected from the reduced number of experiments required when using the quadratic approximation.

8. CONCLUSIONS

The problem of computing a control which minimizes a selected performance measure has been stated and a number of proposed methods for solution have been presented.

Motivation for the use of the gradient algorithm has been discussed and a method of improving the convergence of this procedure utilizing integral mean values of the gradient equation has been derived. Finally, an improved method of determining an approximate gradient step size has been presented.

The selection of:

$$\delta u_j = K_j \int_{t_j}^{t_j+\Delta t} \left[\nabla_u \mathcal{H}(\underline{x}(t), u(t), \underline{p}(t), t) \right] dt$$

is similar to the choice made by Hasdorff and Gupta[11] who applied the method to the solution of sampled data systems.

Further reseach is required in the evaluation of:

$$\frac{\partial}{\partial u_j} F_j(\underline{x}, u, \ldots, u_L, \underline{p}) = \frac{\partial}{\partial u_j} \int_{t_j}^{t_j+\Delta t} \left[\nabla_u \mathcal{H}(\underline{x}(t), u(t), \underline{p}(t), t) \right] dt$$

which may prove useful in determining a more accurate value for δu_j. Efforts in this direction are felt likely to prove fruitful in achieving further improvements in the technique.

REFERENCES

1. Bellman, R. E., Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1957.

2. Bellman, R. E. and Dreyfus, S. E., Applied Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1962.

3. Kirk, D. E., Optimal Control Theory; An Introduction, Ch. 5, to be published.

4. Kelley, H. J., "Gradient Theory of Optimal Flight Paths", Journal of the American Rocket Society, Vol. 30, pp. 947-953, October, 1960.

5. Kelley, H. J., Kopp, R. E. and Moyer, H. G., "A Trajectory Optimization Technique Based Upon the Theory of the Second Variation", Progress in Astronautics and Aeronautics, Vol. 14, 1964.

6. Bryson, A. E. and Denham, W. F., "A Steepest-Ascent Method for Solving Optimum Programming Problems", Journal of Applied Mechanics, Vol. 29, pp. 247-257, June, 1962.

7. Breakwell, J. V., "The Optimization of Trajectories", Journal of the Society of Industrial and Applied Mathematics, Vol. 7, pp. 215-247, June, 1959.

8. McGill, R. and Kenneth, P., "Solution of Variational Problems by Means of a Generalized Newton-Raphson Operator", Journal of the American Institute of Aeronautics and Astronautics, Vol. 2, pp. 1761-1766, October, 1964.

9. Wilde, D. J., Optimum Seeking Methods, pp. 32-35, Prentice-Hall, Inc., Englewood Cliffs, New Jersey,1964.

10. Lapidus, L. and Luus, R., "The Control of Nonlinear Systems", Journal of the American Institute of Chemical Engineers, Vol. 13, pp. 108-113, January, 1967.

11. Hasdorff, L. and Gupta, S.C., "An Iterative Procedure for Optimal Control of a System by Sampled Input", Journal of Electronics and Control, Vol. 16, pp. 177-188, February, 1964.

INITIAL DISTRIBUTION LIST

		No.Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library Naval Postgraduate School Monterey, California 93940	2
3.	Commander-Ships Systems Command Department of the Navy Washington, D.C. 20350	
4.	Commander-Air Systems Command (Code AIR-09B) Department of the Navy Washington, D.C. 20360	1
5.	Professor Donald E. Kirk Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	Professor Harold A. Titus Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
7.	LCDR John A. Battenburg 6683 Maury Drive San Diego, California 92119	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE
An Improved Gradient Algorithm for the Solution of Two-Point Boundary-Value Problems.

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
E. E. Thesis - Electrical Engineering - 1968

5. AUTHOR(S) *(First name, middle initial, last name)*
John Allen Battenburg
Lieutenant Commander, United States Navy

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1968	59	11

8a. CONTRACT OR GRANT NO.

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT
[redacted]

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
Naval Postgraduate School
Monterey, California 93940

13. ABSTRACT

The gradient method for solving two-point boundary-value problems is discussed and a modification of the gradient direction is proposed. An algorithm for efficiently determining the step size is also derived. Analytic and numerical examples illustrating the efficiency of the method are presented.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Optimization						
Computation						
Gradient						
Variational Calculus						
Steepest Ascent						
Search						
Data Processing						
Two-Point Boundary-Value Problems						
Control Systems						